Programming Proofs Project

Fady Adal

December 8, 2023

I thoroughly enjoying working on the project, though I do not think I was successful with it. What started as a hope to formalize the compilation steps of the compiler outlind by Morrisett et al. (1999) ended up as an incomplete formalization of System F using intrinsic types and De Brujin indices. With lots of `admit`s in stuck proofs, this project explored how using De Brujin indices, while theoritically convenient, can become harder to think about in more information-rich scenarios, and the numerous weakening and substitution lemmas become more and more specific. Moreover, it explored how having an intrinically-typed term sounds nice on paper with the inability to construct a type-incorrect term (and thus releaving the need for a preservation proof!) this upfront cost can be hindering. If I were to restart, I would have probably also ended up using intrinsic types and De Brujin indices (especially the former), though.

This project is based heavily on Chapman et al. (2019), but with syntax made to resemble the System F outlined in the Typed Assembly paper. While the former paper gives a full outline of how one formalizes an intrinsically typed System F, they do not show a lot of important lemmas that I had to proof on my own. Moreover, while I found that they're very close, I had to translate different semantics from Agda (the paper) to lean (the project).

Overall, I enjoyed interacting with lean working on this project. I don't think I will abandon this project, since I'm still intrigued about the idea of having a verified compiler to typed assembly (using intrinsic types along the different intermediate calculi), and so will probably work on it during near future.

References

Chapman, J., Kireev, R., Nester, C., and Wadler, P. (2019). System f in agda, for fun and profit. In *International Conference on Mathematics of Program Construction*.

Morrisett, G., Walker, D., Crary, K., and Glew, N. (1999). From system f to typed assembly language. *ACM Trans. Program. Lang. Syst.*, 21(3):527–568.